

Électricité des Hauts-de-France

Prédire la consommation à court terme

1 - Introduction

Souvent, lorsqu'on parle d'estimation en machine learning, il est question de régression. Cependant, dès lors qu'il s'agit de prédire en fonction de la date ou de l'heure, il est nécessaire d'utiliser des technologies adaptées aux données temporelles, dites « time series ». Afin d'en apprendre plus sur ces modèles adaptés utilisables avec Python, j'ai décidé d'en étudier 3 : ARMA et ses dérivés, Prophet et XGBoost.

Pour ce faire, je me suis plongé durant plusieurs jours dans des articles, vidéos et mises en application afin de comprendre leur fonctionnement ainsi que leur intérêt et leurs différences.

L'objectif de cette étude était le suivant: Trouver le modèle le plus adapté à la prédiction de la consommation électrique lors de la semaine suivante. Les données de consommation étant trouvables sur le site de la région, je pouvais ensuite les compléter par des données extérieures qui me semblaient pertinentes (météo, périodes de vacances...).

Lors de l'exploration des données, je me suis rendu compte que nous avions à faire à une série non stationnaire avec deux saisonnalités : Annuelle et Hebdomadaire

Je présenterai donc dans ce rapport, le fruit de mes recherches et expériences en 3 parties. Dans un premier temps, j'expliquerai brièvement quelles sont les différences entre les 3 modèles et je justifierai mon choix, puis le présenterai plus en détail. Je conclurai ensuite et exprimerai quelles sont les notions clés dans le cas de notre problématique.

2 - Les Modèles

A - ARMA et ses dérivés (ARIMA, SARIMA, VARIMAX)

ARMA est certainement le modèle le plus classique et fondamental lorsqu'on parle de modélisation de séries temporelles. Comme son nom l'indique, il repose sur l'auto régression (AR) et la moyenne mobile (MA).

Cependant, sa pertinence repose essentiellement sur les caractéristiques de la série à modéliser, en effet, il faut que celle-ci soit stationnaire afin d'avoir des résultats exploitables. C'est pour cela que des modèles dérivés d'ARMA ont été développés, tel que ARIMA, (I pour Integrated) qui inclue une étape de différenciation intégrée (processus qui consiste en la soustraction des valeurs précédentes) afin d'obtenir une série temporelle stationnaire. Dans le cas d'une série caractérisée par de la saisonnalité, on utilise le modèle SARIMA qui rajoute une composante saisonnière pour prendre en compte les motifs réguliers. Enfin, pour des modélisations plus complexes et dans le cas d'utilisation d'une série temporelle exogène, il est nécessaire d'utiliser le modèle VARIMAX (V pour Vector et X pour eXogeneous). En effet, celui-ci est un modèle dit « multivarié », ce qui nous permet de modéliser plusieurs séries temporelles simultanément afin d'obtenir des résultats toujours plus précis et pertinents.

Dans le cadre de notre recherche, nos données étaient à propos de consommation d'énergie à chaque heure dans chaque région, nous avons remarqué une forte saisonnalité annuelle, c'est pourquoi le modèle SARIMA est le modèle d'auto régression le plus adapté à notre tâche. Cependant, dans le cas d'une recherche approfondie, pour l'utilisation d'une série exogène, comme la météo qui pourrait être pertinente dans notre cas, il serait préférable d'utiliser VARIMAX, afin d'utiliser ces deux séries en parallèle et obtenir des estimations dans le temps plus précises.

Avantages	Inconvénients
Polyvalence (plusieurs modèles très proches)	Difficile de gérer les séries impropres / non linéaires
Rapidité et efficacité dans si la série est stationnaire et suit des saisonnalités simples	Gestion des valeurs manquantes à faire en amont
VARIMAX permet l'utilisation de variables exogènes	

B - Prophet

Prophet est un modèle développé par Facebook, il est prévu pour s'adapter automatiquement aux saisonnalités ainsi qu'à des dates exceptionnelles personnalisables comme les jours fériés ou jours d'événements.

Son mode de fonctionnement est assez simple, il prédit en se basant sur la somme de trois fonctions : tendances, saisonnalité, vacances/jours fériés auxquels s'ajoute une valeur d'erreur.

Son plus gros point fort est son accessibilité, même pour un utilisateur non initié aux modèles de prédiction. En effet, il s'adapte facilement aux valeurs manquantes ou aberrantes, il fournit aussi des intervalles de confiance pour chaque estimation, ce qui permet de facilement évaluer les prédictions.

Cependant, il est très sensible au bruit et aux tendances non linéaires, pour obtenir de bons résultats dans notre recherche, un travail conséquent en amont sera nécessaire pour éliminer le bruit et lisser la série, de plus, il semble impossible d'utiliser des variable exogènes avec Prophet lorsqu'on l'utilise avec darts.

Avantages	Inconvénients
Accessibilité / facilité d'utilisation	Moins adapté à des séries complexes ou très bruitées
Gestion auto des saisonnalités et valeurs manquantes	

C - XGBoost

Le troisième algorithme que j'ai essayé est un algorithme que j'avais déjà utilisé dans un contexte de machine Learning plus classique, il se trouve qu'il est aussi très efficace lorsqu'on travaille avec des séries temporelles.

XGBoost est un algorithme de « gradient boosting », qui ressemble dans son fonctionnement à un système d'arbre de décision, or lorsqu'on parle de boosting, cela signifie que les précédentes itérations de l'arbre effectuées lors de l'entraînement influent sur les prédictions suivantes. et dans le cas du gradient boosting, plusieurs petits arbres de prédiction sont générés en fonction des précédents pour donner au final un grand arbre de prédiction qui servira de modèle de classification ou régression.

C'est un modèle très performant et polyvalent bien qu'un peu compliqué à prendre en main tant les possibilités de réglages sont nombreuses. Il reste plutôt flexible quant à la qualité des données et gère bien les données manquantes ou aberrantes malgré qu'un nettoyage auparavant sera toujours très bénéfique pour les résultats.

Dans notre cas, c'est un modèle qui peut s'avérer très pertinent car il peut répondre à toutes nos problématiques : les deux saisonnalités, la tendance, le bruit et il est aussi possible d'y ajouter des variables exogènes. En revanche, son utilisation dans une démarche de forecast, le paramétrage de xgboost peut être compliqué lorsqu'on n'est pas familier avec l'utilisation de cet outil dans ce contexte.

Avantages	Inconvénients
Modèle réutilisable dans énormément de contexte	Complexité de paramétrage
Excellente performance	
Résistant face aux données bruyantes ou non linéaires	

3 - Conclusion

Au travers de cette veille technologique, j'ai pu découvrir 3 modèles très différents bien qu'ils puissent être utilisés afin de remplir la même mission.

Malgré le côté intimidant du fonctionnement des modèles ARMA, il s'avère que les différents ajouts peuvent répondre sans trop d'efforts à un problème de saisonnalité et de stationnarité et l'ajout de valeur exogène nous permettrait de combler nos données avec la météo.

Prophet, de son côté, excelle dans les estimations à courte durée, avec peu de bruit et des tendances linéaires, à cela s'ajoute une accessibilité imbattable, ce qui le rend idéal pour quiconque souhaitant effectuer des prévisions simples, rapidement et même sans expérience.

Concernant XGBoost, il permet de remplir autant de cases que les modèles ARMA mais ne s'arrête pas aux séries temporelles, ce qui en fait son atout principal, une fois maîtrisé, n'importe quelle régression ou classification peut être effectuée avec une précision et une adaptation bluffante, mais !


4 - Bibliographie / Webographie

ARMA :

- 1.1 / Machine Learning + article, « ARIMA Model – Complete Guide to Time Series Forecasting in Python » by Selva Prabhakaran, <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>
- 1.2 / Medium article, « Time Series Forecasting with ARIMA , SARIMA and SARIMAX » by Brendan Artley, <https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6>
- 1.3 / Youtube videos, « What are ARIMA models » & « What are Seasonal ARIMA models » by Aric LaBarr,

[https://www.youtube.com/playlist?
list=PLjwX9KFWtvNnOc4HtsvaDf1XYG3O5bv5s](https://www.youtube.com/playlist?list=PLjwX9KFWtvNnOc4HtsvaDf1XYG3O5bv5s)

Prophet :

- 2.1 / Kaggle article, « Tutorial: Time Series Forecasting with Prophet » by Prashant Banerjee,
<https://www.kaggle.com/code/prashant111/tutorial-time-series-forecasting-with-prophet>
- 2.2 / Medium article, Time Series Analysis with Facebook Prophet: How it works and How to use it by Mitchell Krieger,
<https://towardsdatascience.com/time-series-analysis-with-facebook-prophet-how-it-works-and-how-to-use-it-f15ecf2c0e3a>
- 2.3 / Kaggle article, «  Time Series forecasting with Prophet » by Rob Mulla,
<https://www.kaggle.com/code/robikscube/time-series-forecasting-with-prophet>

XGBoost :

- 3.1 / Kaggle article, « XGBoost Time series » by Hompi Hump
<https://www.kaggle.com/code/furiousx7/xgboost-time-series>
- 3.2 / Kaggle articles, « Time series forecasting with XGBoost, part 1&2 » by rob mulla
<https://www.kaggle.com/code/robikscube/tutorial-time-series-forecasting-with-xgboost>
<https://www.kaggle.com/code/robikscube/pt2-time-series-forecasting-with-xgboost>